Shallow Semantic Parsing of Randomized Controlled Trial Reports

**Hyung Paek[1], Yacov Kogan[1], Prem Thomas[1], Seymour Codish[1]
and Michael Krauthammer[1,4]**

[1]Center for Medical Informatics, Yale University School of Medicine, New Haven, USA
[4]Department of Pathology, Yale University School of Medicine, New Haven, USA

ABSTRACT

In this work, we are measuring the performance of Propbank-based Machine Learning (ML) for automatically annotating abstracts of Randomized Controlled Trials (RCTs) with semantically meaningful tags. Propbank is a resource of annotated sentences from the Wall Street Journal (WSJ) corpus, and we were interested in assessing performance issues when porting this resource to the medical domain. We compare intra-domain (WSJ/WSJ) with cross-domain (WSJ/medical abstracts) performance. Although the intra-domain performance is superior, we found a reasonable cross-domain performance.

INTRODUCTION

We are interested in semantically annotating abstracts of Randomized Controlled Trials (RCTs). Such reports contain valuable information on medical treatments, their efficacy, side effects and patient population. In order to capture structured information from such reports, it is necessary to annotate the free text with semantically meaningful tags. In this work, we opted for an approach called 'shallow semantic parsing', which is able to dissect sentences into simple WHAT did WHAT TO WHOM, WHEN, WHERE, WHY and HOW. In the case of RCT abstracts, we are interested in extracting results from clinical studies, such as drug trials, where drugs (WHAT$_{DRUG}$) have been tested on some patient population (TO WHOM$_{PATIENT}$) for some specific drug effects (WHAT$_{EFFECT}$). Shallow semantic parsing is usually performed by using machine learning on a semantically annotated training corpus, such as a Propbank, an adjunct to the Penn Treebank [1, 2]. Propbank provides semantic annotation of sentences of the Wall Street Journal Corpus (WSJ), by labeling Penn Treebank constituents with predicate arguments and the roles played by their arguments (see below). In this work, we used Propbank to train classifiers for semantically annotating sentences from medical RCT abstracts. The obvious problem of this approach, the shift from one domain for training (WSJ) to another domain for testing (medical abstracts) has been previously addressed by our group [3]. We found that there is a considerable overlap of verbs, and usages of verbs, between the WSJ/Propbank corpus and abstracts of medical case reports. These findings supported the notion that it is feasible to re-use Propbank (or at least part of it) in the medical domain, enabling the construction of a medical IE system using an existing (albeit non-medical) corpus. In this work, we are trying to *quantify* the performance of Propbank-based shallow semantic parsing on sentences from medical abstracts. Here, we are focusing our efforts on sentences from the "Conclusion" section of RCT abstracts. However, the approach can be easily extended to other types of medical abstracts.

METHODS

Overview: The goal is to automatically label medical sentences with semantic labels corresponding to predicate roles. For example, in a sentence

[A seven week course of pulmonary rehabilitation]$_{Arg0}$ provides [greater benefits]$_{Arg1}$ [to patients]$_{Arg2\text{-}to}$

semantic role labeling recognizes the argument (Arg) positions of the first three roles of the predicate *to provide,* the provider (Arg0), the thing provided (Arg1) and the entity provided for (Arg2), respectively. We are using an existing semantically annotated corpus, called Propbank, to train semantic role classifiers for automating the recognition of the predicate arguments and roles in medical abstracts (for a more detailed description of Propbank, see [1, 2]).

Identification of Predicates: We first explored the types of predicates found in RCT abstracts, and examined the overlap of those predicates with predicates from the WSJ/Propbank corpus. We extracted 10,000 random RCT abstracts that contained an explicit "Conclusion" section. We identified sentences within that section (using the Perl module

Lingua::EN::Sentence), and extracted the sentence predicates by performing a syntax parse –using the Charniak parser [4]- and extracting terminals with VB* POS tags. A final normalization step (using the program morpha [5]) resulted in a list of all normalized verbs, and their frequencies, in the conclusion section of our random abstracts (see Table 1 for the top 10 verbs).

Table 1. The 10 most frequent verbs in RCT abstracts

#	*Occurences*	*Verb*	*Cumulative frequency*
1	1238	reduce	0.036
2	1163	improve	0.070
3	1056	suggest	0.100
4	963	increase	0.129
5	888	use	0.155
6	808	associate	0.178
7	742	compare	0.200
8	733	show	0.221
9	718	provide	0.242
10	593	appear	0.260

Similar to the results obtained previously [3] we found that the top 10 verbs covered >25%, and the top 100 verbs covered >71% of all verb occurrences in our sample set of RCT abstracts. Also, 99 of those 100 verbs are annotated in Propbank. This finding reinforces the notion that we can re-use Propbank for semantic role labeling in the medical domain. However, we must be aware of the fact that these 99 verbs are not used in exactly the same way in both the WSJ corpus and medical abstracts. In our previous study, we found that the usages – at least for high-frequency verbs - seem to be quite consistent across domains [3], but listed important exceptions from this rule. For example, the predicate *to diagnose* exhibits a particular use in the medical domain, such as in diagnosing a symptom as a particular disease. For that particular example, it is important to capture the predicate sense for assigning the correct predicate roles. Some verbs may exhibit consistent domain-dependent verb usage. For example, *to discharge* is a polysemous verb that is mostly used in the context of releasing a patient from the hospital in the medical domain. In this work, we assumed consistent verb usage within and across domains. We will discuss the implication of this assumption below.

Experimental setup: We decided to train semantic role classifiers for the top 5 verbs in our RCT corpus (Table 1). We ran different experiments for evaluating the performance of our approach. We report on the following two experimental setups:

1. Training and testing on the WSJ/Propbank corpus (within domain)
2. Training on WSJ/Propbank corpus and testing on RCT abstracts (cross-domain)

The first setup allows for validation of our Machine Learning (ML) approach, of the choice of ML features and algorithm. The second setup is used to assess the ML performance across domains. It should be noted that the second setup includes the use of automated syntax parsing (unlike the first setup, which is solely based on manual syntax information from Penn Treebank).

The use of automated syntax parsing generally results in measured performance drops. This is because automated parsing is usually less accurate than manual ones.

Feature Extraction for Machine Learning (ML): In the following paragraphs, we discuss the background and specifics for extracting syntactic features for semantic role labeling via ML. This is best approached by looking at the problem as a straightforward task of finding binary semantic labels for each word in a medical sentence. Let's look at a sentence containing the predicate *to reduce*.

The article discussed that [Zocor]$_{\text{Arg0}}$ reduced [cholesterol]$_{\text{Arg1}}$ [in the intervention group]$_{\text{ArgM-in}}$.

In Propbank, Arg0 represents the *Agent* (or thing that causes the reduction), and Arg1 represents the *thing reduced*, while ArgM represents a modifier arguments (*where* the reduction occurred). A ML classifier will evaluate all features of a word (such as *Zocor*), and decide whether it is Arg0 of the verb *reduce* or not. Another classifier will assess the probability that Zocor is Arg1 of reduce, and so forth. A final classifier will assess the probability that a word is not an argument (NULL) of the predicate (such as *article* in the sample sentence above). The final label is assigned according to the highest classifier probability. The result of the ML learning step are classifiers for each of the verb arguments, and the NULL case. These classifiers are generic, and can be applied to any of our five target verbs. This is made possible by using the names of the five predicates as features in the ML algorithm.

ML classifiers make their decision based on features, in this case the features of the individual words in the medical sentences. We are using non-lexical features that have been previously discussed as being useful in semantic role labeling [6]. We are limiting ourselves to non-lexical features as we are operating across domains (WSJ and medical corpora) with quite distinct lexical contents. The features can be derived from the syntax parse tree of the sentences:

```
(S (NP (DT The) (NN article))
    (VP (VBD discussed)
     (SBAR (IN that)
      (S (NP (NNP Zocor))
       (VP (VBD reduced)
        (NP (NN cholesterol))
        (PP (IN in) (NP (DT the) (NN intervention) (NN group)))))))
    (. .))
```

The parse tree contains constituents, such as noun phrases (NP) and verb phrases (VP), as well as terminals/POS tags, such as determiners (DT) or nouns (NN). For each word in the sentence, we are extracting features from the parse trees, in order to classify the words according to its correct semantic role. This approach is called W-by-W (word by word) semantic role labeling. However, there is another approach called C-by-C (constituent by constituent) semantic role labeling. C-by-C does not look at individual words in the sentences, but at the tree constituents corresponding to the semantic roles of the predicates. For example, the PP constituent (prepositional phrase) in the parse tree above corresponds to ArgM (*in the intervention group*) of the sentence predicate. The task is to decide for each constituent whether it represents a predicate argument or not. The disadvantage of the C-by-C approach is the possibility of conflicting constituent labeling, with the result of multiple role assignments to the corresponding terminals/words. We decided to use a W-by-W approach, while retaining features of the C-by-C approach. Based on the observation in Propbank that most constituents corresponding to predicate arguments are 1 or 2 nodes above the respective words/terminals, we calculated word features from the syntactic properties of the constituents 1 and 2 levels above those words. Here is an illustrative example: For the word *Zocor,* we calculated features of both the NP and S constituent in the parse tree (1 and 2 nodes above the terminal). We used the following features (features with a * indicate static features, which are the same across all words in the sentence):

1. *Predicate – We are using the predicate of the sentences as a feature
2. Path – the syntactic path from a word to the sentence predicate. (For the word *Zocor* - more specifically: it's non-terminal constituent NP and S - the paths are NP↑S↑VP↓VBD and S↑VP↓VBD).
3. The Phrase Type: The syntactic category of the constituent (NP and S for *Zocor*).
4. Position: the position of the word relative to the predicate. (The word *Zocor* is before the predicate)
5. Head Word POS: The POS tag of the syntactic head of the constituent (NNP for the *Zocor's* first constituent NN, and NP for the S constituent).
6. *Subcategorization: The rule expanding the predicate's parent node. (VP→VBD-NP for the predicate *reduce*)

Training and Testing Sets: For Setups 1 and 2 (intra-domain and cross-domain), we prepared the following testing and training sets:

Setup 1: We extracted sentences from the Propbank/WSJ corpus containing our five target predicates [the five verbs that occurred most frequently in RCT abstracts, see Table 1]. Ignoring words that did not constitute core arguments (such as modifier arguments ArgMs) resulted in a corpus of 15,424 annotated words. We used 12,500 of these as a training set, from which we constructed classifiers for core arguments Arg0, Arg1…Arg4, as well as NULL (i.e., words that do not constitute predicate arguments). As mentioned above, we did not prepare a training set for each of the five predicates separately, as the predicate is used as a feature itself. We tested the resulting classifiers on the remaining 2,924 words from our 15,424 word corpus.

Setup 2: For Setup 2, we used the same argument classifiers as discussed above, which have been trained on 12,500 words from the Propbank/WSJ corpus. However, we tested these classifiers on a set of annotated medical sentences. Two medical experts independently looked at 250 predicate sentences from RCT abstracts that contained the five target verbs, and manually labeled each word in those sentences with the corresponding predicate role. This resulted in a (medical) testing set of 6,373 words.

Machine Learning Architecture: We used the SVMTorch[1] program for ML. SVMTorch is a Support Vector Machine (SVM) program that allows for multi-class learning. We ran the program using the Gaussian kernel, with all other parameters set to the default values.

[1] http://www.idiap.ch/machine_learning.php?content=Torch/en_SVMTorch.txt

RESULTS AND DISCUSSION

In Table 2, we show the results from training and testing on the WSJ/Propbank corpus (Setup 1). We calculated the recall (R), precision (P) and F1 measures – 2RP/(R+P) – for each argument (including the NULL argument) separately. We achieve F score measures between .52 and .81 among the 5 predicate arguments, and .86 for the NULL argument. SVMTorch also provides a multiclass misclassification error of .19 In other words, the ML classifier correctly classifies the predicate argument for approximately 80% of the words in the testing set.

Table 2 Intra-domain classification performance

Arg	*Recall*	*Precision*	*F*	*n*
NULL	0.84	0.86	0.86	1574
0	0.55	0.48	0.52	236
1	0.85	0.76	0.81	936
2	0.93	0.45	0.61	152
3	0.00	0.00	N/A	9
4	0.78	0.64	0.71	17

Table 3 Cross-domain classification performance

Arg	*Recall*	*Precision*	*F*	*n*
NULL	0.81	0.70	0.75	3351
0	0.72	0.33	0.45	745
1	0.67	0.86	0.75	1952
2	0.60	0.24	0.34	325
3				0
4				0

Table 3 shows the results of training on words from the WSJ/Probank corpus, and testing on words from medical RCT abstracts (as can bee seen, we did not encounter predicates with more than three arguments). The performance is consistently below the performance of the intra-domain situation (Table 2). The multiclass misclassification error is .29. In other words, the Propbank-trained classifier assigns the correct predicate argument in approximately 70% of the words in medical RCT abstracts.

It should be noted that our performance numbers do not necessarily compare to similar numbers reported in earlier publications on semantic role labeling. Such studies often include the identification of the predicate argument *span* in the calculation of precision and recall. In contrast, we ignored the argument span and looked at each word separately, which may result in higher performance numbers. However, the main purpose of our study was assessing the *difference* in intra-domain and cross-domain ML performance, and we believe that our experimental design is sufficient for this purpose. We find a reasonable cross-domain performance, with a 10% drop in multiclass classification accuracy.

The lower accuracy can be explained as follows: First, we are using an automated syntax parse in the cross-domain setup. Second, our annotation strategy of the medical corpus may be different from the one used in the Propbank/WSJ corpus. This seems apparent in the case of Argument 2 (table 3), where we record a high number of false positives (low precision). Apparently, the markup of the medical corpus was rather conservative with regard to Argument 2. Third, we did not take into account verb usage differences. While this may not be a major problem with any of the five predicates used in our experiments, we are aware of verbs that are used differently within and cross-domain corpora (see discussion above). Propbank provides labels for the specific verb usage in the WSJ corpus. We are thinking of ways to include this information as an additional ML feature.

In our experiments, we opted for exclusion of lexical features, as we expected them to be rather unique in the respective domain corpora. As a consequence, we achieve a rather low intra- (and cross-) domain performance for arguments that would normally profit from the availability of those features. For example, we achieve low F scores for Arg0, which often signifies the verb subject. Lexical information, such as the word '*physician*', has a strong probability of being associated with Arg0. The importance of lexical features is an important reason to consider the construction of an annotated medical training corpus. However, we can envision ways to improve the ML performance with additional non-lexical features. For example, we could use *semantic* features that are consistent across the WSJ and medical domain, such as generic named entities (persons, location etc.).

We would like to mention another challenge that we encountered when performing our study: the detection of the 'Conclusion' section within RCT abstracts. We relied on explicit mentioning of the word *conclusion* in those abstracts. However, in many instances, authors do not use any explicit demarcation of the conclusion section. In order to circumvent this problem, there is the possibility to use zoning of medical abstracts, as discussed previously [7].

ACKNOWLEDGEMENTS

We are thankful for Nigel Collier's helpful comments. This project has been partially funded by NIH grant 5R01DA012690-07

1. Kipper, K., M. Palmer, and O. Rambow. Extending PropBank with VerbNet Semantic Predicates. Workshop on Applied Interlinguas AMTA 2002. 2002. Tiburon, CA.
2. Kingsbury, P., M. Palmer, and M. Marcus. Adding Semantic Annotation to the Penn TreeBank. Human Language Technology Conference. 2002. San Diego, CA.
3. Kogan, Y., N. Collier, S. Pakhomov, and M. Krauthammer. Towards Semantic Role Labeling and IE in the Medical Domain. Proc AMIA Symp, 2005.
4. Charniak, E., A Maximum-Entropy-Inspired Parser. 1999, Brown University.
5. Minnen, G., J. Carroll, and D. Pearce. Applied morphological processing of English. Natural Language Engineering, 2001. 7(3): p. 207-223.
6. Pradhan, S., W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. HLT/NAACL. 2004. Boston.
7. McKnight, L. and P. Srinivasan, Categorization of sentence types in medical abstracts. AMIA Annu Symp Proc, 2003: p. 440-4.